



International Timetabling Competition 2011: An Adaptive Large Neighborhood Search algorithm

Sørensen, Matias; Kristiansen, Simon; Stidsen, Thomas Riis

Published in:

Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)

Publication date:

2012

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Sørensen, M., Kristiansen, S., & Stidsen, T. R. (2012). International Timetabling Competition 2011: An Adaptive Large Neighborhood Search algorithm. In *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)* (pp. 489-492)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

International Timetabling Competition 2011: An Adaptive Large Neighborhood Search algorithm

Matias Sørensen · Simon Kristiansen ·
Thomas R. Stidsen

Keywords Adaptive Large Neighborhood Search · High School Timetabling ·
International Timetabling Competition 2011 · XHSTT

1 Introduction

An algorithm based on Adaptive Large Neighborhood Search (ALNS) for solving the generalized High School Timetabling problem in XHSTT-format (Post et al (2012a)) is presented. This algorithm was among the finalists of round 2 of the International Timetabling Competition 2011 (ITC2011). For problem description and results we refer to Post et al (2012b).

2 Adaptive Large Neighborhood Search

Adaptive Large Neighborhood Search was first developed as a metaheuristic for the class of Vehicle Routing Problems (Pisinger and Ropke (2005); Ropke and Pisinger (2006)). It has been applied for few other problem classes as well, including Project Scheduling (Muller (2009, 2010)), Lot-sizing (Muller and Spoorendonk (2011)), Optimal Statistic Median Problem (Katterbauer et al (2012)).

Recently we have developed a framework based on ALNS for solving combinatorial optimization problems (written in C# 4.0). This framework is part of the commercial product Lectio¹, where it is used to solve various practical timetabling problems, see Kristiansen et al (2011); Sørensen and Stidsen (2012) and Kristiansen and Stidsen (2012).

The pseudo code for a general ALNS algorithm is given in Algorithm 1.

Matias Sørensen [msso@dtu.dk], Simon Kristiansen, Thomas R. Stidsen
Operations Research, DTU Management Engineering
Technical University of Denmark

¹ <http://www.lectio.dk>
Cloud-based administration system for high schools. Developed by MaCom A/S, Vesterbro-
gade 48 1., 1620 Copenhagen V, Denmark

Algorithm 1 Adaptive Large Neighborhood Search

```

1: candidate solution  $x$ , remove-methods  $\Omega^-$ , insert-methods  $\Omega^+$ 
2:  $x_{\text{best}} = x$ 
3: while stop-criterion not met do
4:    $x' = x$ 
5:   RemoveStrategy: select  $q$  as some quantity to be removed
6:   AdaptiveStrategy: select remove-method  $r \in \Omega^-$  and insert-method  $i \in \Omega^+$ 
7:   remove requests from  $x'$  using  $r(q)$ 
8:   insert requests into  $x'$  using  $i$ 
9:   AdaptiveStrategy: update performance indicators
10:  if  $c(x') \leq c(x_{\text{best}})$  then
11:     $x_{\text{best}} = x'$ 
12:  end if
13:  AcceptStrategy: set candidate solution  $x$  to either  $x'$ ,  $x_{\text{best}}$  or  $x$  itself
14: end while
15: return  $x_{\text{best}}$ 

```

The main points of the algorithm are described below in general terms.

- In each iteration, a remove and insertion method is chosen and applied to the candidate solution. The combination of these methods defines the neighborhood of the algorithm, hence there exists $|\Omega^-| \cdot |\Omega^+|$ different neighborhoods.
- **RemoveStrategy**: Governs the selection of q . This has major influence on how much computational time each iteration requires.
- **AdaptiveStrategy**: Responsible for selecting remove and insertion methods in each iteration, and updating their respective performance indicators of these method by some metric.
- **AcceptStrategy**: Determines which solution to use as candidate solution for next iteration. This could in principle be any known solution, but is usually selected as either the current candidate solution x itself, the newly produced solution x' , or the current best solution x_{best} .

3 Algorithm setup for ITC2011

Here we describe our implementation of a ALNS algorithm for the XHSTT format. The choice of ALNS strategies are briefly mentioned below. More details will be available in the full paper.

- **RemoveStrategy**: The remove and insertion methods deal with sub-events. q is defined as the sum of the duration of the sub-events which are removed from the solution. We select q as a random number, bounded by a percentage of the total duration of all instance events.
- **AdaptiveStrategy**: We have chosen a metric essentially based on two parameters for each method; The number of times the method was part of an iteration which yielded a better solution than the current one, and the

relative gap between the current solution and the resulting solution from applying the method.

- **AcceptStrategy:** An acceptance criteria borrowed from Simulated Annealing (SA) is used, with the following additional property: If no new best solution has been found in a number of iterations, the temperature is increased by a factor, and the candidate solution is set to the best known solution. The intention is to allow more diverse exploring of the area around the best known solution, in case the algorithm gets 'stuck'.

Let a *move* be a small perturbation on a solution. The following moves are used in this implementation: Move $M_{se,t}$ denotes the assigning of sub-event se to time t . $M_{r,er,se}$ denotes the assigning of resource r to event resource er on sub-event se . Furthermore we also implement the corresponding unassign-moves, denoted $M_{se,t}^-$ and $M_{r,er,se}^-$, respectively.

Using these moves a total of 9 insertion methods (all more or less based on the greedy principle, e.g. regret heuristics (Potvin and Rousseau (1993); Sørensen and Stidsen (2012)), and 14 remove methods (all based on some element of relatedness and an element of randomness) are implemented. These methods are divided into three categories, based on what they (un-)assign: Only times, only resources, or both times and resources.

An example of a remove method is the following, which removes sub-events from non-preferred times: Given an XHSTT instance, and a solution S to this instance. Find all tuples $\langle se, t \rangle$ of S , where sub-event se is assigned time t , and t is not a preferred time for sub-event se (see *Prefer times constraints*, Kingston (2010)). Let the set of these tuples be denoted U . Select randomly a subset of these tuples $\bar{U} \subseteq U$ such that the sum of the duration of all sub-events of the tuples in \bar{U} equals q . Perform an unassign time move $M_{se,t}^-$ for each of the tuples in \bar{U} .

An example of an insertion method is the following: Let $\Delta(M) \in \mathbb{R}$ be the profit of performing move M on the solution at hand S . Select $M_{\text{best}} = \arg \min_{se,t} (\Delta(M_{se,t}))$, and if $\Delta(M_{\text{best}}) \leq 0$, apply M_{best} to S and repeat, otherwise stop. This is a greedy method which assigns times to sub-events, until no profitable move can be found.

In the full paper all insert/remove methods will be described in detail.

The final algorithm contains 9 free parameters, which were tuned for best performance using the *irace* package (see López-Ibáñez et al (2011); Birattari (2005)).

4 Final remarks

This paper documents how Adaptive Large Neighborhood Search can be applied to problems in XHSTT format.

The proposed algorithm was applied to all instances in archive XHSTT-ITC2011, and showed competitive results in most cases (comparing to the best known solutions at that point in time).

ALNS has not been used much in the field of timetabling, but we see no reason to believe that ALNS should not perform well on other (related) problems in this field.

Acknowledgements Thank you goes to Michael Herold for fruitful discussions concerning ALNS strategies. Thank you to Manuel López-Ibáñez for help using the `irace` package. And finally thank you to David Pisinger for advice on ALNS implementation.

References

- Birattari M (2005) The Problem of Tuning Metaheuristics as seen from a Machine Learning Perspective, vol 292 *Dissertations in Artificial Intelligence - Infix*, 1st edn. Springer
- Katterbauer K, Oguz C, Salman S (2012) Hybrid adaptive large neighborhood search for the optimal statistic median problem. *Computers & Operations Research* 39(11):2679 – 2687
- Kingston JH (2010) The hseval high school timetable evaluator. URL <http://it.usyd.edu.au/~jeff/hseval.cgi>
- Kristiansen S, Stidsen TR (2012) Adaptive large neighborhood search for student sectioning at danish high schools. In: *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*
- Kristiansen S, Sørensen M, Stidsen T, Herold M (2011) Adaptive large neighborhood search for the consultation timetabling problem, to appear
- López-Ibáñez M, Dubois-Lacoste J, Stützle T, Birattari M (2011) The irace package: Iterated racing for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, Université Libre de Bruxelles, IRIDIA, Av F. D. Roosevelt 50, CP 194/6 1050 Bruxelles, Belgium, <http://iridia.ulb.ac.be/irace>
- Muller L (2009) An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. In: *MIC 2009: The VIII Metaheuristics International Conference*
- Muller L (2010) An adaptive large neighborhood search algorithm for the multi-mode resource-constrained project scheduling problem. I, Department of Management Engineering, Technical University of Denmark Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark
- Muller L, Spoorendonk S (2011) A hybrid adaptive large neighborhood search algorithm applied to a lot-sizing problem. *European Journal of Operational Research* Volume 218(Issue 3):614–623
- Pisinger D, Ropke S (2005) A general heuristic for vehicle routing problems. *Computers & Operations Research* 34:2403–2435
- Post G, Ahmadi S, Daskalaki S, Kingston J, Kyngas J, Nurmi C, Ranson D (2012a) An xml format for benchmarks in high school timetabling. *Annals of Operations Research* 194:385–397
- Post G, Gaspero LD, Kingston JH, McCollum B, Schaerf A (2012b) The third international timetabling competition. In: *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, Son, Norway
- Potvin JY, Rousseau JM (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research* 66(3):331 – 340
- Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40:455–472
- Sørensen M, Stidsen TR (2012) High school timetabling: Modeling and solving a large number of cases in denmark. In: *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*